UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/892,296 | 06/26/2001 | David Eatough | 3408.2.16 | 2361 |

21552 7590 07/23/2008

MADSON & AUSTIN
15 WEST SOUTH TEMPLE
SUITE 900
SALT LAKE CITY, UT 84101

| EXAMINER | |
|---|---|
| CHOUDHURY, AZIZUL Q | |
| ART UNIT | PAPER NUMBER |
| 2145 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 07/23/2008 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/892,296 | EATOUGH ET AL. | |
| | Examiner | Art Unit | |
| | AZIZUL CHOUDHURY | 2145 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *03 April 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 June 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

***Detailed Action***

This office action is in response to the correspondence received on April 3, 2008.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Farinacci et al (US Pat No: 5,519,704) in view of Tseung (US Pat No: 5,036,518), hereafter referred to as Farinacci and Tseung, respectively.

1. As to Claims 1, 4, and 7, Farinacci teaches through Tseung: Receiving, by a server computer, a request to perform a task for a plurality of computers over a network (column 5, lines 50-53, Farinacci), wherein the task comprises installing a software application or updating a software application (column 40, lines 51-66, Tseung); Performing said task using a multicast message communicated from said server computer over said network, wherein the performance of said task is triggered by an event occurring on said server computer or said network (column 5, lines 55-57, Farinacci); Updating a task status table by said server, wherein said task status table indicates whether said task has been completed for each said plurality of computers; Receiving, by said server computer, a request to

complete said task from a first computer (see column 5, lines 53-55, Farinacci); Prioritizing requests to complete said task; if more than one request is received by said server computer from each of a plurality of computers (see column 48, lines 41-54, Tseung); Determining whether said task was completed for said first computer using said task status table (see column 5, lines line 60-63, Farinacci); Performing said task using a unicast message communicated from said server computer over said network to said first computer in accordance with said determination (see column 5, lines 64-67, Farinacci) and said prioritization (see column 48, lines 41-54, Tseung); and Updating said task status table, indicating whether said task has been completed for said first computer

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the tasks software updates and installs and doesn't teach the status table and also does not teach the tasks being triggered by events. Farinacci is also silent regarding prioritizing requests. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). In addition, means by which to maintain the status of tasks in a computing device that is handling tasks is obvious and well known in the art. Tseung teaches how the retransmission station maintains data structures (table) to keep track of the status of messages (tasks or program

transmissions) to different recipients (Figure 40 and column 18, lines 16-47, Tseung). For instance, it can record if there are crc errors. When no errors are left, it is known that the messages have been transmitted completely and correctly (column 36, line 21- column 37, line 15, Tseung). Plus, Tseung teaches how events trigger tasks as claimed (see column 18, lines 59-67; column 19, lines 21-56; column 22, lines 19-32; column 23, line 53 – column 24, line 19; and column 26, line 32 – column 27, line 5; Tseung) Finally, Tseung also teaches how when requests from more than one computer are received, the requests are serialized (equivalent to prioritizing requests); see column 48, lines 41-54, Tseung. The serializing of requests is a prioritizing of requests based on time. Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

2. As to Claims 2, 5 and 13, Farinacci teaches through Tseung: Wherein said determining whether said task was completed for said first computer comprises: Receiving an identifier for said first computer; Searching a task status table using said identifier; Retrieving a status indicator associated with said identifier; and Determining whether said task was completed for said first computer using said status indicator (see column 2, lines 57-63, Farinacci).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the task being software updates and installs. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

3. As to Claims 3, 6, 8, and 11, Farinacci teaches through Tseung: Wherein said receiving said request to complete said task from said first computer comprises: Determining whether said first computer is in communication with said network; and Sending said request to complete said task from said first computer (see column 53-55, Farinacci).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the task being software updates and installs. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure

also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

4. As to Claim 9, Farinacci teaches through Tseung: A storage medium: Said storage medium including stored instructions that, when executed by a processor, result in receiving, by a server computer, a request to perform a task for a plurality of devices over a network (see column 5, lines 50-53, Farinacci), performing said task using a multicast message communicated from said server computer over said network, wherein the performance of said task is triggered by an event occurring on said server computer or said network (see column 5, lines 55-57, Farinacci), receiving, by said server computer, a request to complete said task from at least one device (see column 5, lines 53-55, Farinacci), prioritizing requests to complete said task, if more than one request is received by said server computer from each of a plurality of devices (see column 48, lines 41-54, Tseung), determining whether said task was completed for said at least one device, and performing said task using a unicast message communicated from said server computer over said network to said at least one device in accordance with said determination (see column 5, lines 60-67, Farinacci) and said

prioritization (see column 48, lines 41-54, Tseung), wherein the task comprises copying a file, installing a software application or updating a software application (column 40, lines 31-66, Tseung)).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the tasks of software updates and installs and doesn't teach the status table and also does not teach the tasks being triggered by events. Farinacci is also silent regarding prioritizing requests. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). In addition, means by which to maintain the status of tasks in a computing device that is handling tasks is obvious and well known in the art. Tseung teaches how the retransmission station maintains data structures (table) to keep track of the status of messages (tasks or program transmissions) to different recipients (Figure 40 and column 18, lines 16-47, Tseung). For instance, it can record if there are crc errors. When no errors are left, it is known that the messages have been transmitted completely and correctly (column 36, line 21- column 37, line 15, Tseung). Plus, Tseung teaches how events trigger tasks as claimed (see column 18, lines 59-67; column 19, lines 21-56; column 22, lines 19-32; column 23, line 53 – column 24, line 19; and column 26, line 32 – column 27, line 5; Tseung) Finally, Tseung also teaches

how when requests from more than one computer are received, the requests are serialized (equivalent to prioritizing requests); see column 48, lines 41-54, Tseung. The serializing of requests is a prioritizing of requests based on time. Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

5. As to Claim 10, Farinacci teaches through Tseung: Wherein the stored instructions, when executed by a processor, further result in determining whether said task was completed for said at least one device by receiving an identifier for said at least one device, searching a task status table using said identifier, retrieving a status indicator associated with said identifier, and determining whether said task was completed for said at least one device using said status indicator (see column 2, lines 57-63, Farinacci).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the task being software updates and installs. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). Therefore, it would have been

obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

6. As to Claim 12, Farinacci teaches through Tseung: A storage medium; Said storage medium including stored instructions that, when executed by a processor, result in receiving, by a server computer, a request to send information to a plurality of devices (see column 5, lines 50-53, Farinacci), sending said information, from said server computer, to said plurality of devices using a broadcast message, wherein sending of said information is triggered by an event occurring on said server computer or said network (see column 5, lines 55-57, Farinacci), receiving a request for said information from at least one device (see column 5, lines 53-55, Farinacci), prioritizing requests for said information, if more than one request is received by said server computer from each of a plurality of devices (see column 48, lines 41-54, Tseung), determining whether said at least one device received said information, and sending said information, from said server computer, to said at least one device using a unicast message in accordance with said determination, and updating said task status table, wherein said task status table comprises a status indicator indicating whether said information has been received by said at least one device (see column 5, lines 60-67, Farinacci).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the tasks of software updates and installs and doesn't teach the status table and also does not teach the tasks being triggered by events. Farinacci is also silent regarding prioritizing requests. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). In addition, means by which to maintain the status of tasks in a computing device that is handling tasks is obvious and well known in the art. Tseung teaches how the retransmission station maintains data structures (table) to keep track of the status of messages (tasks or program transmissions) to different recipients (Figure 40 and column 18, lines 16-47, Tseung). For instance, it can record if there are crc errors. When no errors are left, it is known that the messages have been transmitted completely and correctly (column 36, line 21- column 37, line 15, Tseung). Plus, Tseung teaches how events trigger tasks as claimed (see column 18, lines 59-67; column 19, lines 21-56; column 22, lines 19-32; column 23, line 53 – column 24, line 19; and column 26, line 32 – column 27, line 5; Tseung). Finally, Tseung also teaches how when requests from more than one computer are received, the requests are serialized (equivalent to prioritizing requests); see column 48, lines 41-54, Tseung. The serializing of requests is a prioritizing of requests based on time.

Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

7. As to Claim 14, Farinacci teaches through Tseung: Wherein the stored instructions, when executed by a processor, further result in receiving a request for said information by connecting said at least one device to said network and sending said request for said information from said at least one device (see column 5, lines 60-67, Farinacci).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the task being software updates and installs. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

8. As to Claim 15, Farinacci teaches through Tseung: A storage medium; said storage medium including stored instructions that, when executed by a processor, result in receiving, by a server computer, a request to perform a task for a plurality of devices over a network (see column 5, lines 50-53, Farinacci), performing said task using a multicast message communicated from said server computer over said network, wherein the performance of said task is triggered by an event occurring on said server computer or said network (see column 5, lines 55-57, Farinacci), receiving, by said server computer, a request to complete said task from at least one device (see column 53-55, Farinacci), prioritizing requests to complete said task, if more than one request is received by said server computer from each of a plurality of devices (see column 48, lines 41-54, Tseung), searching a task status table using an identifier, retrieving a status indicator associated with said identifier, determining whether said task was completed for said at least one device using said status indicator (see column 2, lines 57-63, Farinacci), and performing said task using a unicast message communicated from said server computer over said network to said at least one device in accordance with said determination (see column 5, lines 60-67, Farinacci) and said prioritization (see column 48, lines 41-54, Tseung), wherein the task comprises installing a software application or updating a software application (column 40, lines 31-66, Tseung).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the tasks

of software updates and installs and doesn't teach the status table and also does not teach the tasks being triggered by events. Farinacci is also silent regarding prioritizing requests. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). In addition, means by which to maintain the status of tasks in a computing device that is handling tasks is obvious and well known in the art. Tseung teaches how the retransmission station maintains data structures (table) to keep track of the status of messages (tasks or program transmissions) to different recipients (Figure 40 and column 18, lines 16-47, Tseung). For instance, it can record if there are crc errors. When no errors are left, it is known that the messages have been transmitted completely and correctly (column 36, line 21- column 37, line 15, Tseung). Plus, Tseung teaches how events trigger tasks as claimed (see column 18, lines 59-67; column 19, lines 21-56; column 22, lines 19-32; column 23, line 53 – column 24, line 19; and column 26, line 32 – column 27, line 5; Tseung) Finally, Tseung also teaches how when requests from more than one computer are received, the requests are serialized (equivalent to prioritizing requests); see column 48, lines 41-54, Tseung. The serializing of requests is a prioritizing of requests based on time. Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung

to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

9. As to Claim 16, Farinacci teaches through Tseung: Wherein the stored instructions, when executed by a processor, further result in receiving said request to complete said task from at least one device by connecting said at least one device to said network, and sending said request to complete said task from said at least one device (see column 5, lines 60-67, Farinacci).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the task being software updates and installs. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

10. As to Claim 17, Farinacci teaches through Tseung: A server, said server having a task manager module to manage complete of a task for a plurality of target

devices using a multicast message communicated from said server, wherein the task comprises installing a software application or updating a software application and wherein performance of said task is triggered by an event occurring on said server (column 40, lines 51-66, Tseung); a plurality of target devices, said plurality of target devices each having a task finisher module to request completion of said task if uncompleted, wherein said requests to complete said task are prioritized by said task manager module (see column 48, lines 41-54, Tseung), and wherein the task finisher module is configured to install or update applications; and A network to communicate information between said server and said plurality of target devices to complete said task (see column 4, lines 40-47, Farinacci).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the tasks of software updates and installs and doesn't teach the status table and also does not teach the tasks being triggered by events. Farinacci is also silent regarding prioritizing requests. In the same field of endeavor, Tseung teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). In addition, means by which to maintain the status of tasks in a computing device that is handling tasks is obvious and well known in the art. Tseung teaches how the retransmission station maintains data

structures (table) to keep track of the status of messages (tasks or program transmissions) to different recipients (Figure 40 and column 18, lines 16-47, Tseung). For instance, it can record if there are crc errors. When no errors are left, it is known that the messages have been transmitted completely and correctly (column 36, line 21- column 37, line 15, Tseung). Plus, Tseung teaches how events trigger tasks as claimed (see column 18, lines 59-67; column 19, lines 21-56; column 22, lines 19-32; column 23, line 53 – column 24, line 19; and column 26, line 32 – column 27, line 5; Tseung). Finally, Tseung also teaches how when requests from more than one computer are received, the requests are serialized (equivalent to prioritizing requests); see column 48, lines 41-54, Tseung. The serializing of requests is a prioritizing of requests based on time. Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

11. As to Claim 18, Farinacci teaches through Tseung: Further comprising a task handler module for each of said plurality of target devices to complete said task for said plurality of target devices (see column 4, lines 40-47, Farinacci).

(While Farinacci teaches a design allowing for tasks to be performed through the use of unicast and multicast messages, Farinacci does not disclose the task being software updates and installs. In the same field of endeavor, Tseung

teaches a design allowing for software installs and updates through multicasts (column 33, lines 60-62 and column 40, lines 51-66, Tseung). The disclosure also teaches how one-to-one (unicast) data transfers are allowed (column 1, lines 25-58 and column 40, lines 31-51, Tseung). Therefore, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Farinacci with those of Tseung to allow software and/or updates to be sent using the guaranteed, reliable and secure one-to-many technique (column 40, lines 51-54, Tseung)).

12. The obviousness motivation applied to claims 1, 4, 7, 9, 12, 15 and 17 are applicable to their respective dependent claims.

***Response to Remarks***

The amendment received on April 3, 2008 has been carefully examined but is not deemed fully persuasive. In lieu of the claim amendments, the 112-type rejection has been overcome and hence that rejection is withdrawn. The 103-type rejection using the Farinacci and Tseung prior arts continue to stand though. The current amendment features independent claim amendments to add the feature of, "prioritizing requests to complete said task, if more than one request is received by said server computer from each of a plurality of computers." The applicant contends that neither prior art teach this claim limitation, the examiner disagrees. Tseung teaches how when requests from more than one computer are received, the requests are serialized (equivalent to

prioritizing requests); see column 48, lines 41-54, Tseung. The serializing of requests is a prioritizing of requests based on time. The claims themselves fail to specify the prioritizing is based on any factor other than time, hence the Tseung art's teachings of serialization are deemed equivalent to the claimed limitation of prioritization.

***Conclusion***

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to AZIZUL CHOUDHURY whose telephone number is (571)272-3909. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jason Cardone can be reached on (571) 272-3933. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/A. C./
Examiner, Art Unit 2145

/Jason D Cardone/
Supervisory Patent Examiner, Art Unit 2145